Computing Machinery I
Assignment 2
8% of your final score
Due October 9th @ 11:59PM MST

Lead TA
The lead TA for this assignment is Abdelghani Guerbas

Objective
The objective of this assignment is to practice branching and looping in ARMv8 assembly.

Skills Needed for this Assignment
- Ability to work with basic arithmetic, loops, and if-else constructs in assembly
- Ability to print to standard output using the `printf()` and `scanf()` functions
- Ability to optimize assembly code by rearranging loops and using alternate instructions
- Ability to use macros in assembly code
- Ability to assemble programs using *gcc* and use *m4* to process macros
- Ability to use *gdb* to debug and display assembly language programs

Overview
Your program will generate N random numbers between 0 and 9 where each number represents the number of occurrences of a word in a text document. The N words are all in one single document. Your program will compute the frequency of each word in the document and displays the highest and lowest frequencies.

Details
The program prompts the user for N, an integer between 5 and 20, generates N random positive integers between 0 and 9, and then displays these numbers and the size of the document. Your program will then compute the frequency of each word using the following formula and displays the highest and lowest frequencies.

Frequency of a word = (100*number of occurrences) / size of the document.

The size of the document is the sum of occurrences of all of its words (sum of all N numbers).

For reading N, you will need to call the C library scanf() function.

Write two versions of the program:

1. Write the program without macros (i.e. don't use *m4*). Use a pre-test loop, where the test is at the top of the loop.
2. Rewrite the above program by putting the loop test at the bottom of the loop (make sure it is still a pretest loop) and add macros to the above program to make it more readable (use *m4*). In particular, provide macros for heavily used registers.

Make sure your code is properly formatted into columns, is readable and fully documented, and includes identifying information at the top of each file. You must comment each line of assembly code. Your code should also be well designed: make sure it is well organized, clear, and concise.

Submission

- **Note:** The lead TA may provide further submission instructions
- Name your programs *assign2a.s* (no macros) and *assign2b.asm* (with macros)
- Create a script file for each version. Call them *assign2a.script* and *assign2b.script* • Submit a *README* file providing extra instructions or information for your TA (optional)
- Submit your work to the appropriate dropbox on D2L.

Late Submission Policy

Late submissions will be penalized as follows:
-12.5% for each late day or portion of a day for the first two days
-25% for each additional day or portion of a day after the first two days
Hence, no submissions will be accepted after 5 days (including weekend days) of the announced deadline.

Academic Misconduct

This assignment is to be done by individual students: your final submission must be your own original work. Teamwork is not allowed. Any similarities between submissions will be further investigated for academic misconduct. While you are encouraged to discuss the assignment with your colleagues, this must be limited to conceptual and design decisions. Code sharing by any means is prohibited, including *looking* at someone else's paper or screen. The submission of compiler generated assembly code is absolutely prohibited. Any re-used code of excess of 5 lines in C and 10 lines in assembly (10 assembly language instructions) must be cited and have its source acknowledged. Failure to credit the source will also result in a misconduct investigation.

D2L Marks

Marks posted on D2L are subject to change (up or down).

Computing Machinery I
Assignment 2 Rubric

Student:____

Item	**Max Points**	**Points**	**Comments**
Code compiles	5		
Code runs	5		
Calculating document size	10		
Calculating highest frequency	15		
Calculating lowest frequency	15		
Random number generation	15		
User interface & input validation	10		
Code readability (formatting and documentation)	10		
Two versions of a pretest loop	5		
Macros in version b	5		
Script files	5		
Total Points	**100**		